



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/099,849	03/15/2002	Barry M. Baker	POU901138US1	5701

46369 7590 01/10/2005

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 COLUMBIA CIRCLE
ALBANY, NY 12203

EXAMINER

CHOW, CHIH CHING

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 01/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/099,849	Applicant(s) BAKER ET AL.	
	Examiner Chih-Ching Chow	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 March 2002.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-51 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-51 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 15 March 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>03/15/2002</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the application filed on March 15, 2002.
2. The priority date considered for this application is March 15, 2002.
3. Claims 1-51 have been examined.

Specification

4. The disclosure is objected to because of the following informalities:

paragraphs 0002 and 0003, Ser. Numbers are missing. Appropriate correction is required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-3, 5, 6, 9-12, 17-19, 21, 22, 25-28, 33, 34, 36-38, 40, 41, and 44-47 are rejected under 35 U.S.C. 102(b) as being anticipated by Keith Besaw et al. US. Patent No. 5,555,412 (hereinafter "Besaw").

CLAIM

1. A method of facilitating the use of aliases in debugging of applications, said method comprising:

- a. obtaining an alias of an application to be debugged, the alias being unknown to at least a debug generation stage of a compiler of the application; and
- b. resolving the alias name known to the compiler to facilitate use of the alias in debugging of the application.

2. The method of claim 1, wherein said resolving comprises using an alias table to determine the name known to the compiler.

B saw

Besaw teaches a method of facilitating the use of alias in debugging, see Besaw column 7, lines 31-32, "the invention is used to simplify handling of a **debug boundary**". For item a, see Besaw column 2, lines 20-22, "It would also be beneficial to have the flexibility to **add compiler generated variables (known to compiler)** that are **aliased (unknown to compiler)** to user variables to provide special views of user variables". For item b, in Besaw column 7, lines 36-37, "A pseudo variable (*alias*) can be generated and added to the check-list of every variable affected by the debug boundary (*the implementation can be a 'table', e.g. an 'alias table'*). Then, at each debug boundary, simply mark the pseudo variable as defined." (*resolving the alias name known to the compiler*) Also in Besaw's abstract, "The pseudo variable is attached to the check-list of each **variable** included in the predefined **alias class**."

For the feature of claim 1 see claim 1 rejection. As to matching alias to the name known to the compiler, see Besaw's claim 1, "constructing a respective pseudo variable to represent each said **predefined large alias class (alias table)** in the source program and, for each **variable (known to the compiler)** included within each said predefined large alias class attaching said

respectively pseudo variable to said corresponding check-lists".

3. The method of claim 2, wherein said alias table comprises one or more entries, and wherein an entry of the one or more entries includes the alias and the name known to the compiler corresponding to the alias.

For the feature of claim 2 see claim 2 rejection. Besaw's alias class (*alias table*) implies there can be one or more entries.

5. The method of claim 2, wherein said alias table is dynamically maintainable.

For the feature of claim 2 see claim 2 rejection. Besaw's alias class (*alias table*) is dynamically maintainable based on the currently running source program.

6. method claim 2, wherein the alias is added to the alias table at runtime.

For the feature of claim 2 see claim 2 rejection. Besaw's alias class (*alias table*) is added at runtime based on the currently running source program.

9. The method of claim 1, wherein said obtaining comprises receiving expression from a user debugging the application, said expression including said alias.

For the feature of claim 1 see claim 1 rejection. Besaw's disclosure teaches receiving expression from a user, and the expression is an alias, see Besaw column 1, lines 39-41, "if two or more expressions denote the same memory address, the expressions are aliases of one another."

10. A method of facilitating debugging of applications, said method comprising:
a. obtaining an application to be debugged, the alias lacking associated debug information; and
b. debugging at least a portion of the application using the alias.

In Besaw's disclosure the alias class only contains the alias information, not the debugging information. The alias information is used for debugging, however it has to do 'inverse checking' to connect with the debugging information. See Besaw column 2 lines

29-31, "to provide an aliasing method and apparatus using a combination of direct effect and inverse checking that improves the efficiency of code optimization and provides more flexibility than methods used in the past which rely exclusively on a direct effect apparatus."

11. The method of claim 10, wherein the debugging comprises:
 resolving the alias to a formal name the alias; and
 obtaining debug information the formal name.

For the feature of claim 10 see claim 10 rejection. For the rest of the claim 11 feature see claim 1 and 10 rejections.

12. The method of claim 11, wherein said resolving comprises using an alias table to determine the formal name.

For the feature of claim 11 see claim 11 rejection. For the rest of the claim 12 feature see claim 1, 2, and 10 rejections.

17. A system of facilitating the use of aliases debugging of applications, said system comprising:
 a. an alias of an application to be debugged, the alias being unknown to least a debug generation stage of a compiler of the application; and
 b. means for resolving the alias to a name known to the compiler to facilitate use of the alias in debugging the application.

Same as claim 1 rejection, see Besaw's claim 11, his invention is also for a digital computer, therefore it applies for a 'system'.

18. The system of claim 17, wherein said means for resolving comprises means using an alias table to determine the

Same as claim 17 rejection.

name known to the compiler.

19. The system of claim 18, wherein said alias table comprises one or more entries, and wherein an entry of the one or more entries includes the alias and the name known to the compiler corresponding to the alias.

For the feature of claim 18 see claim 18 rejection. For the rest of the claim 19 feature see claim 1 and 3 rejections.

21. The system of claim 18, wherein said alias table is dynamically maintainable.

For the feature of claim 18 see claim 18 rejection. For the rest of the claim 21 feature see claim 5 rejection.

22. The system of claim 18, wherein the alias is added to the alias table at runtime.

For the feature of claim 18 see claim 18 rejection. For the rest of the claim 22 feature see claim 6 rejection.

25. The system of claim 17, further comprising means for receiving an expression from a user debugging the application, said expression including said alias.

For the feature of claim 17 see claim 17 rejection. For rest of the features see claim 9 rejection.

26. A system of facilitating debugging of applications, said system comprising:
a. an alias of an application to be debugged, the alias lacking associated debug information; and
b. means for debugging at least a portion of the application using the alias.

Same as claim 10 rejection.

27. The system of claim 26, wherein the means for debugging comprises:
a. means for resolving the alias to a formal name of the alias; and

For the feature of claim 26 see claim 26 rejection. For the rest of claim 27 feature see claim 1 and 10 rejections.

b. means for obtaining debug information for the formal name.

28. The system claim 27, wherein said means for resolving comprises means for using an alias table to determine the formal name.

For the feature of claim 27 see claim 27 rejection. For the rest of claim 28 feature see claim 10 rejection.

33. A system of facilitating the use of aliases in debugging of applications, said system comprising:

Same as claim 1 rejection.

- a. an alias of an application to be debugged, the alias being unknown to at least a debug generation stage a compiler of the application; and
- b. a debugger to resolve the alias name known to the compiler to facilitate use of the alias in debugging of the application.

34. A system of facilitating debugging of applications, said system comprising:

Same as claim 1 and 10 rejections.

- a. an alias of an application to be debugged, the alias lacking associated debug information; and
- b. a debugger to debug at least a portion of the application using the alias.

36. At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of facilitating the use of aliases in debugging of applications, said method comprising:

Same as claim 1 rejection, see Besaw's claim 11, his invention is also for a digital computer, therefore it includes at least one 'program storage device'.

- a. obtaining an alias of an application to

be debugged, the alias being unknown to least a debug generation stage of a compiler of the application; and

b. resolving the alias a name known to the compiler to facilitate use of the alias in debugging of the application.

37. The at least one program storage device of claim 36, wherein said resolving comprises using an alias table to determine the name known to the compiler.

For the feature of claim 36 see claim 36 rejection. For the rest of claim 37 feature see claim 1 rejection.

38 The at least one program storage device of claim 37, wherein said alias table comprises one or more entries, and wherein an entry of one or more entries includes the alias and the name known to the compiler corresponding to the alias.

For the feature of claim 37 see claim 37 rejection. For the rest of claim 38 feature see claim 1 and 3 rejections.

40. The at least one program storage device of claim 37, wherein said alias table is dynamically maintainable.

For the feature of claim 37 see claim 37 rejection. For the rest of claim 40 feature see claim 5 rejection.

41. The at least one program storage device of claim 37, wherein the alias is added to the alias table at runtime.

For the feature of claim 37 see claim 37 rejection. For the rest of claim 41 feature see claim 6 rejection.

44. The at least one program storage device of claim 36, wherein said obtaining comprises receiving an expression from a user debugging the application, said expression including said alias.

For the feature of claim 36 see claim 36 rejection. For the rest of claim 44 feature see claim 9 rejection.

45. At least one program storage device

Same as claim 10 rejection.

readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of facilitating debugging of applications, said method comprising:

- a. obtaining an alias of an application to be debugged, the alias lacking associated debug information; and
- b. debugging at least a portion using the alias.

46. The least one program storage device of claim 45, wherein the debugging comprises:

- a. resolving the alias to a formal name of the alias; and
- b. obtaining debug information for the formal name.

For the feature of claim 45 see claim 45 rejection. For the rest of claim 46 feature see claim 1 and 2 rejections.

47. The least one program storage device of claim 46, wherein said resolving comprises using an alias table to determine the formal name.

For the feature of claim 46 see claim 46 rejection. For the rest of claim 47 feature see claim 1 and 2 rejections.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2122

8. Claims 4, 7, 8, 13-16, 20, 23, 24, 29-32, 35, 39, 42, 43, and 48-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,555,412 by Keith Besaw et al. (hereinafter "Besaw"), in view of U.S. Patent No. 6,324,683 by You-Chin Fuh et al. (hereinafter "Fuh").

CLAIM

4. The method of claim 3, wherein the entry further includes an indication of one or more users authorized to use said alias.

Besaw / Fuh

For the feature of claim 3 see claim 3 rejection. Besaw teaches all aspects of claim 4, but he does not mention 'authorized users' specifically, however, Fuh teaches it in an analogous prior art. In Fuh's column 31, lines 45-49, "Also included is the information necessary to **obtain authorization for the debugger** to attach a monitor/controller to the user's program (i.e., login ID and password) as well as the instruction address in the user's program where the debugging session should begin", and column 7, lines 66-67, under 'Authorization': "For security purposes, the DBMS process usually runs under a special UID so that normal users cannot attach to it." In column 24, lines 43-46, "since **debugging** activity is controlled by DBMS, **authority checking** currently adopted in controlling database operations can be easily extended to **control the debugging requests.**" It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Besaw's disclosure of the creating alias

class (*alias table*) for debugging by using authorized user(s) taught by Fuh, for the purpose of providing enhanced security (Fuh, column 2, lines 63).

7. The method claim 2, wherein said alias table comprises one entry having the alias and indication of one or more users authorized use the alias, and another entry having the alias and an indication of one or more other users authorized to use the alias.

For the feature of claim 2 see claim 2 rejection. In Fuh's column 9, lines 49-52, "**Authorization control over the debugging requests** from the applications is provided by the **authority checking function** of the DBMS" - it implies one or more users are authorized to access the data.

8. The method of claim 1, wherein said resolving comprises determining whether a user debugging the application is authorized to use alias, and resolving the alias in response to the determining indicating authorization.

For the feature of claim 1 see claim 1 rejection. For the rest of the claim 8 feature see claim 4 rejection.

13. A method of managing the use of aliases in debugging of applications, said method comprising:

See claim 1 and 4 rejections.

- a. determining whether a user attempting to use an alias in debugging at least a portion of an application is authorized to use the alias; and
- b. providing debug information for the alias, in response to the determining indicating authorization.

14. The method of claim 13, wherein said determining comprises checking an alias table to determine whether the user is authorized.

For the feature of claim 13 see claim 13 rejection. Even Besaw does not teach including the authorized user in his alias class (*alias table*), but it would have

been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Besaw's disclosure of the creating alias class (*alias table*) for debugging by using authorized user(s) taught by Fuh, for the purpose of providing enhanced security (Fuh, column 2, lines 63).

15. The method of claim 13, wherein said providing comprises resolving the alias to a name known to a compiler of the application in order to associate the debug information of that name with the alias.

For the feature of claim 13 see claim 13 rejection. For the rest of the claim 15 feature see claim 1 and 10 rejections.

16. The method of claim 15, wherein said resolving comprises using an alias table to determine the name known to the compiler.

For the feature of claim 15 see claim 15 rejection. For the rest of the claim 16 feature see claim 1 and 10 rejections.

20. The system of claim 19, wherein the entry further includes an indication of one or more users authorized to use said alias.

For the feature of claim 19 see claim 19 rejection. For the rest of claim 20 feature see claim 4 and 7 rejections.

23. The system of claim 18, wherein said alias table comprises one entry having the alias and an indication of one or more users authorized to use the alias, and another entry having the alias and an indication of one or more other users authorized to use the alias.

For the feature of claim 18 see claim 18 rejection. For the rest of claim 23 feature see claim 7 rejection.

24. The system of claim 17, wherein said means for resolving comprises means for

For the feature of claim 17 see claim 17 rejection. For the rest of claim 24

determining whether a user debugging the application is authorized to use the alias, and resolving the alias in response to the determining indicating authorization.

feature see claim 4 rejection.

29. A system of managing the use of aliases in debugging of applications, said system comprising:

For item a, see claim 1 rejection; for item b, see claim 4 rejection.

a. means for determining whether a user attempting to use an alias in debugging at least a portion of an application is authorized to use the alias; and

b. means for providing debug information for the alias, in response to the determining indicating authorization.

30. The system of claim 29, wherein said means for determining comprises means for checking an alias table to determine whether the user is authorized.

For the feature of claim 29 see claim 29 rejection. For the rest of claim 30 feature see claim 4 rejection.

31. The system of claim 29, wherein said means for providing comprises means for resolving the alias to a name known compiler of the application in order to associate the debug information that name with the alias.

For the feature of claim 29 see claim 29 rejection. For the rest of claim 31 feature see claim 4 and claim 10 rejections.

32. The system of claim 31, wherein said means for resolving comprises means for using an alias table to determine the name known to the compiler.

For the feature of claim 31 see claim 31 rejection. For the rest of claim 32 feature see claim 1 and 10 rejections.

35. A system of managing the use of aliases in debugging of applications, said system comprising:

- a. a debugger determine whether a user attempting to use an alias debugging at least a portion of an application is authorized to use the alias; and
- b. the debugger to provide debug information for the alias, in response to the determining indicating authorization.

Same as claim 4 rejection.

39. The at least one program storage device of claim 38, wherein the entry further includes an indication of one or more users authorized to use said alias.

For the feature of claim 38 see claim 38 rejection. For the rest of claim 39 feature see claim 7 rejection.

42. The at least one program storage device of claim 37, wherein said alias table comprises one entry having the alias and an indication of one or more users authorized to use the alias, and another entry having the alias and an indication of one or more other users authorized to use the alias.

For the feature of claim 37 see claim 37 rejection. For the rest of claim 42 feature see claim 4 rejection.

43. The least one program storage device of claim 36, wherein said resolving comprises determining whether a user debugging the application is authorized to use the alias, and resolving the alias in response to the determining indicating authorization.

For the feature of claim 36 see claim 36 rejection. For the rest of claim 43 feature see claim 4 rejection.

48. At least one program storage device

See claim 4 and 36 rejections.

readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing the use of aliases in debugging of applications, said method comprising:

- a. determining whether a user attempting to use an alias in debugging at least a portion of an application is authorized to use the alias; and
- b. providing debug information for the alias, in response to the determining indicating authorization.

49. The at least one program storage device of claim 48, wherein said determining comprises checking an alias table to determine whether the user authorized.

For the feature of claim 48 see claim 48 rejection. For the rest of claim 49 feature see claim 4 rejection.

50. The at least one program storage device of claim 48, wherein said providing comprises resolving the alias to a name known to a compiler of the application in order to associate the debug information of that name with the alias.

For the feature of claim 48 see claim 48 rejection. For the rest of claim 50 feature see claim 2, 4 and 10 rejections.

51. The at least one program storage device of claim 50, wherein said resolving comprises using an alias table to determine the name known to the compiler.

For the feature of claim 50 see claim 50 rejection. For the rest of claim 50 feature see claim 1 rejections.

Art Unit: 2122

Conclusion

The following summarizes the status of the claims:

35 USC § 102 claim rejections: 1-3, 5, 6, 9-12, 17-19, 21, 22, 25-28, 33, 34, 36-38, 40, 41, and 44-47.

35 USC § 103 claim rejections: 4, 7, 8, 13-16, 20, 23, 24, 29-32, 35, 39, 42, 43, and 48-51.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:00am - 3:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow

Examiner

Art Unit 2122

cc 

**ANTHONY NGUYEN-BA
PRIMARY EXAMINER**